

Платежный API

Формат запросов и ответов

По стандарту Open Banking

Open Banking — это предоставление доступа к банковским данным третьим лицам, по согласию пользователей.

Успешные ответы возвращаются со статусом 200 OK/201 Created в формате:

```
{
  "data": {
    ...
  },
  "risk": {},
  "meta": {},
  "links": {
    "self": "...
  }
}
```

Описание атрибутов:

| Значение | Описание |
|----------|---|
| data | Содержит полезную нагрузку эндпоинта |
| risk | Содержит информацию о рисках (актуально только для API платежей, в остальных процессах возвращается в виде пустого объекта) |
| meta | Содержит различную мета-информацию, например, количество доступных страниц для запросов с пейджингом |
| links | Содержатся ссылки на запрошенную сущность и связанные с ней сущности |

Возможные поля links:

| Значение | Описание |
|------------|---|
| links.self | URL созданной/запрошенной сущности. Всегда возвращается |
| links.next | Если существует следующая страница записей ресурсов, то ППУ предоставляет ссылку на следующую страницу ресурсов в поле ответа |

| | |
|-------------|---|
| links.prev | Если предыдущая страница записей ресурсов существует, то ППУ предоставляет ссылку на предыдущую страницу ресурсов в поле ответа. Отсутствие предыдущей ссылки указывает на то, что текущая страница является первой страницей результатов |
| links.first | Ссылку на первую страницу результатов в поле |
| links.last | Ссылку на последнюю страницу результатов в поле |

Ошибки возвращаются со статусом 400 Bad Request/404 Not Found/500 Internal Server Error/403 Forbidden в формате:

```
{
  "code": "InvalidRequest",
  "id": "b1bb4834-1196-49b5-9ea9-705d8d420bae",
  "message": "Некорректный запрос",
  "errors": [
    {
      "errorCode": "RU.CBR.Field.Invalid",
      "message": "Список разрешений не содержит минимального
необходимого набор разрешений",
      "path": "Data.Permissions"
    }
  ]
}
```

Описание атрибутов:

| Значение | Описание |
|--------------|---|
| code | Содержит высокоуровневый код ошибки |
| id | Идентификатор инцидента внутри Бланка в виде UUID |
| message | Высокоуровневое описание ошибки |
| errors | Содержит список детализированных ошибок |
| errorCode | Содержит код ошибки по стандарту ЦБ |
| errorMessage | Содержит описание ошибки |
| path | Содержит путь к свойству в теле запроса, которое вызвало ошибку |

Используемые высокоуровневые коды ошибок:

| Значение | Описание |
|---------------------|----------------------------|
| InvalidRequest | Некорректный запрос |
| InvalidToken | Некорректный токен доступа |
| InternalServerError | Внутренняя ошибка |

Используемые коды ошибок:

| Значение | Описание |
|--------------------------------------|--|
| RU.Blanc.Access.Denied | Попытка обратиться к данным, к которым у клиента нет доступа |
| RU.CBR.Field.Expected | Не передано значение указанного поля |
| RU.CBR.Field.Invalid | В указанном поле передано некорректное значение |
| RU.CBR.Field.InvalidDate | Передана некорректная дата |
| RU.CBR.Field.Missing | Указанное поле не передано |
| RU.CBR.Resource.ConsentMismatch | Запрошенный ресурс не соответствует переданному согласию |
| RU.CBR.Resource.InvalidConsentStatus | Некорректный статус согласия (согласие не авторизовано клиентом) |
| RU.CBR.Resource.InvalidFormat | Некорректный формат модели |
| RU.CBR.Rules.AfterCutoffDateTime | Попытка запросить ресурс после окончания срока его действия |
| RU.CBR.Unsupported.AccountIdentifier | Передан неподдерживаемый индикатор счета |
| RU.CBR.UnexpectedError | Произошла непредвиденная внутренняя ошибка на стороне банка |
| RU.CBR.Header.Invalid | Передано некорректное значение в указанном заголовке |
| RU.CBR.Header.Missing | Указанный заголовок не был передан |
| RU.CBR.Resource.NotCreated | Попытка обратиться к ещё не созданному ресурсу |
| RU.CBR.Rules.ResourceAlreadyExists | Попытка повторно создать существующий ресурс |
| RU.CBR.Signature.Invalid | Некорректная подпись запроса |
| RU.CBR.Signature.InvalidClaim | В подписи передано некорректное утверждение |
| RU.CBR.Signature.MissingClaim | В подписи не передано необходимое утверждение |
| RU.CBR.Signature.Malformed | Подпись искажена |
| RU.CBR.Signature.Missing | Отсутствует подпись |
| RU.CBR.Unsupported.AccountIdentifier | Идентификатор счёта не поддерживается указанной схемой |
| RU.CBR.Unsupported.LocalInstrument | Указанный локальный инструмент не поддерживается банком |
| RU.CBR.Reauthenticate | Для продолжения необходимо заново аутентифицировать клиента |

Не рекомендуется показывать переданные описания ошибок клиентам, т. к. они ориентированы больше для разработчиков сторонних сервисов. Для клиентов лучше формировать собственные тексты ошибок в зависимости от переданного кода ошибки.

Для нестандартных запросов

Успешные ответы возвращаются со статусом 200 OK в формате:

```
{
  "value": {
    ...
  },
  "isSuccess": true,
  "error": null
}
```

| Значение | Описание |
|-----------|---|
| value | Содержит непосредственно полезную нагрузку ответа, структура определяется конкретным запросом |
| isSuccess | Успешное выполнение запроса |
| error | Ошибка при выполнении запроса |

Ошибки возвращаются со статусом 400 Bad Request в формате:

```
{
  "value": null,
  "isSuccess": false,
  "error": {
    "type": "InvalidPhoneNumber",
    "message": "Некорректный формат номера телефона"
  }
}
```

Используемые коды ошибок:

| Значение | Описание |
|--------------|--------------------------|
| errorType | Строковый код ошибки |
| errorMessage | Содержит описание ошибки |

Не рекомендуется показывать эти описания клиентам, т. к. они ориентированы больше для разработчиков сторонних сервисов. Для клиентов лучше формировать собственные тексты ошибок в зависимости от переданного кода ошибки.

Пояснения по некоторым часто используемым полям

Поля DebtorParty И CreditorParty

Идентифицируют счёт отправителя и получателя соответственно.

```
{
  "schemeName": "RU.CBR.BBAN",
  "identification": "77777777777777777777"
}
```

Описание атрибутов:

| Значение | Описание |
|----------------|---|
| schemeName | Описывает способ идентификации счёта |
| identification | Передаёт идентификатор счёта, соответствующий указанной схеме |
| name | Название счёта. Поле необязательное |

Поля DebtorParty И CreditorParty

Идентифицируют сторону отправителя и получателя соответственно.

```
{
  "inn": "7717777777",
  "name": "АО Рога и Копыта",
  "kpp": "7727777777"
}
```

Описание атрибутов:

| Значение | Описание |
|----------|----------------------|
| name | Название контрагента |
| inn | ИНН контрагента |
| kpp | КПП контрагента |

Поля DebtorAgent И CreditorAgent

Идентифицируют банк отправителя и получателя соответственно.

```
{
  "schemeName": "RU.CBR.BIK",
  "identification": "7777777777",
  "name": "ООО «Бланк банк»"
}
```

Описание атрибутов:

| Значение | Описание |
|----------------|---|
| schemeName | Описывает способ идентификации банка |
| identification | Передаёт идентификатор счёта, соответствующий указанной схеме |
| name | Название банка. Необязательное поле |
| PostalAddress | Адрес контрагента. Необязательное сложное поле |

Поле TaxInformation

Содержит данные налоговых платежей.

```
{
  "uin": "0",
  "payerStatus": "13",
  "kbk": "1806666866600666666",
  "oktmo": "64658686",
  "paymentReason": "0",
  "period": "ГД.00.2020",
  "documentNumber": "0",
  "documentDate": "0"
}
```

Описание атрибутов:

| Значение | Описание |
|----------------|---|
| uin | УИН (22) |
| payerStatus | Статус плательщика (101) |
| kbk | КБК (104) |
| oktmo | ОКТМО (105) |
| paymentReason | Основание платежа (106) |
| period | Налоговый период (107) |
| documentNumber | Номер документа (108) |
| documentDate | Дата документа (109). Формат ДД.ММ.ГГГГ |

Прочие замечания

- все DateTime-переменные передаются в UTC
- bookingDateTime – время создания транзакции
- valueDateTime – время исполнения платежа

Аутентификация

Общая информация

Аутентификация в рамках Open Banking происходит по стандартам OIDC (OpenID Connect). Все запросы выполняются только от имени клиента банка, m2m-взаимодействия не предусмотрено.

Определены такие скоупы API:

1. `open-banking:accounts` — доступ к счетам (информация по счёту, балансы, выписки, списки операций)
2. `open-banking:payments` — создание и отправка платежей
3. `open-banking:offline_access` — доступ к ресурсам от имени пользователя на длительное время
4. `open-banking:openid` — проверка подлинности для авторизации доступа к сведениям о пользователе

После аутентификации клиента и получения пары токенов (`access` и `refresh`), сторонний сервис должен передавать `access`-токен в каждом запросе в бланковский API в `Authorization header`:

```
Authorization: Bearer {{accessToken}}
```

В нашей реализации доступны 2 флоу аутентификации:

- стандартный OIDC'овский `authorization_code`
- кастомный `open_banking:third_party`

Флоу `authorization_code`

Стандартный флоу аутентификации по OIDC. Подразумевает редирект на форму аутентификации Бланка (`identity.blanc.ru`):

```
GET /api/sso/connect/authorize?  
client_id={clientId}&  
response_type=code&  
scope=openid offline_access open-banking:accounts open-banking:payments&  
nonce=<nonce>&  
state=<state>&  
code_challenge=<code_challenge>&  
code_challenge_method=<code_challenge_method>&  
redirect_url=<redirect_url>
```

Флоу open_banking:third_party

Кастомный флоу аутентификации. Подразумевает использование форм аутентификации на стороне партнёрского приложения, т. е. без редиректов на страницы Бланка.

Флоу подразумевает прохождение клиентом трёх шагов:

1. ввод номера телефона – после этого клиенту отправляется одноразовый код для подтверждения
2. ввод одноразового кода
3. ввод пароля

После этого сторонний сервис получает временный `exchange_token`, который запросом на стандартный OIDC'овский эндпоинт `connect/token` можно обменять на полноценную пару `access-` и `refresh-` токенов.

Разберём последовательно.

Шаг 1. Ввод номера телефона

Сторонний сервис отображает клиенту форму ввода номера телефона, после заполнения которого отправляет запрос:

```
POST /api/sso/v2/open-banking/auth/third-party/login
Content-Type: application/json

{
  "ClientId": "my.service.id",
  "PhoneNumber": "799999999999"
}
```

Описание атрибутов:

| Значение | Описание |
|--------------------------|---|
| <code>clientId</code> | Это идентификатор, выданный Бланком партнёрскому сервису |
| <code>PhoneNumber</code> | Номер телефона, введенный клиентом. Вводится с семеркой впереди |

Ответ возвращается в формате:

```
{
  "value": {
    "confirmToken": "...",
    "payload": {
      "codeTtlSeconds": 59
    }
  },
  "isSuccess": true,
  "error": null
}
```


Описание атрибутов:

| Значение | Описание |
|--------------|---|
| confirmToken | Ограниченный по времени действия токен, который нужно передать на следующем шаге аутентификации |
| payload | Содержит дополнительную информацию |

Значения параметров компоненты payload:

| Значение | Описание |
|----------------|---|
| codeTtlSeconds | Передаётся срок действия одноразового кода в секундах |

После обработки этого запроса Бланк отправляет клиенту одноразовый код, который он должен ввести в форме на стороннем сервисе.

Ошибка возвращается в формате:

```
{
  "value": null,
  "isSuccess": false,
  "error": {
    "type": "InvalidPhoneFormat",
    "message": "Некорректный формат номера телефона"
  }
}
```

Возможные коды ошибок на этом шаге:

| Значение | Описание |
|--------------------|--|
| InvalidClient | стороннему приложению не доступен API Open Banking в целом или флоу open_banking:third_party в частности |
| InvalidPhoneFormat | Некорректный формат номера телефона |
| InvalidUser | Пользователю не доступен API Open Banking по той или иной причине (например, не является клиентом Бланка) |
| ServiceUnavailable | Произошла ошибка при отправке одноразового кода (например, недоступен СМС-провайдер), необходимо повторить попытку позднее |

Шаг 2. Ввод одноразового кода

Сторонний сервис отображает клиенту форму ввода одноразового кода, после заполнения которого отправляет запрос:

```
POST /api/sso/v2/open-banking/auth/third-party/verifyCode
Content-Type: application/json

{
  "ClientId": "my.client.id",
  "ConfirmToken": "...",
  "Code": "1234"
}
```

Описание атрибутов:

| Значение | Описание |
|--------------|--|
| confirmToken | Передаётся временный токен, выданный на предыдущем шаге |
| code | Передаётся введённый клиентом код (четырёхзначное число) |

Успешный ответ возвращается в формате:

```
{
  "value": {
    "confirmToken": "...",
  },
  "isSuccess": true,
  "error": null
}
```

Ошибка возвращается в формате:

```
{
  "value": null,
  "isSuccess": false,
  "error": {
    "type": "InvalidConfirmToken",
    "message": "Некорректный токен подтверждения"
  }
}
```

Возможные коды ошибок на этом шаге:

| Значение | Описание |
|---------------------|--|
| InvalidConfirmToken | Передан неверный confirmToken |
| InvalidUser | Пользователь не найден или не имеет доступа к API Open Banking |
| CodeExpired | Срок действия одноразового кода истёк |
| CodeVerifyThrottled | Слишком много неудачных попыток ввести код, нужно подождать |
| InvalidCode | Неверный код |

Шаг 3. Ввод пароля

Сторонний сервис отображает клиенту форму ввода пароля, после заполнения которого отправляет запрос:

```
POST /api/sso/v2/open-banking/auth/third-party/2fa
Content-Type: application/json

{
  "ClientId": "my.client.id",
  "ConfirmToken": "...",
  "Password": "qwerty123"
}
```

Описание атрибутов:

| Значение | Описание |
|--------------|---|
| confirmToken | Передаётся временный токен, выданный на предыдущем шаге |
| password | Передаётся введённый клиентом пароль |

Успешный ответ возвращается в формате:

```
{
  "value": {
    "exchangeToken": "...",
  },
  "isSuccess": true,
  "error": null
}
```

exchangeToken содержит одноразовый токен с ограниченным сроком действия, который сторонний сервис должен обменять на пару полноценных токенов на следующем шаге.

Ошибка возвращается в формате:

```
{
  "value": null,
  "isSuccess": false,
  "error": {
    "type": "InvalidConfirmToken",
    "message": "Некорректный токен подтверждения"
  }
}
```

Возможные коды ошибок на этом шаге:

| Значение | Описание |
|---------------------|--|
| InvalidConfirmToken | Передан неверный confirmToken |
| InvalidUser | Пользователь не найден или не имеет доступа к API Open Banking |
| InvalidPassword | Неверный пароль |

Шаг 4. Получение пары токенов

После успешного прохождения всех форм клиентом сторонний сервис отправляет запрос по стандарту OIDC:

```
POST /api/sso/connect/token
Content-Type: application/x-www-form-urlencoded

client_id=my.client.id&
scope=offline_access openid open-banking:payments open-banking:accounts&
grant_type=open_banking:third_party&
exchange token=...
```

В `exchange_token` передаётся одноразовый токен, полученный на предыдущем шаге.

Успешный ответ возвращается в формате:

```
{
  "access_token": "...",
  "expires_in": 3600,
  "token_type": "Bearer",
  "refresh_token": "...",
  "scope": "offline_access open-banking:accounts open-banking:payments
openid"
}
```

Описание атрибутов:

| Значение | Описание |
|--------------|---|
| accessToken | лежит полноценный JWT для доступа к API |
| refreshToken | токен обновления |

Ошибка возвращается в формате:

```
{
  "error": "invalid_request",
  "error_description": "invalid_request"
}
```

error содержит код ошибки по стандарту OIDC.

Обновление токенов

Токены обновляются по стандарту OIDC запросом:

```
POST /api/sso/connect/token
Content-Type: application/x-www-form-urlencoded

client_id=my.client.id&
grant_type=refresh_token&
refresh_token=...
```

Успешный ответ возвращается в формате:

```
{
  "id_token": "...",
  "expires_in": 3600,
  "token_type": "Bearer",
  "refresh_token": "...",
  "scope": "open-banking:accounts open-banking:payments openid
offline_access"
}
```

Ошибка возвращается в формате:

```
{
  "error": "invalid_grant"
}
```

error содержит код ошибки по стандарту OIDC.

Предоставление доступа к счетам

6.2.1.2. Диаграмма последовательности

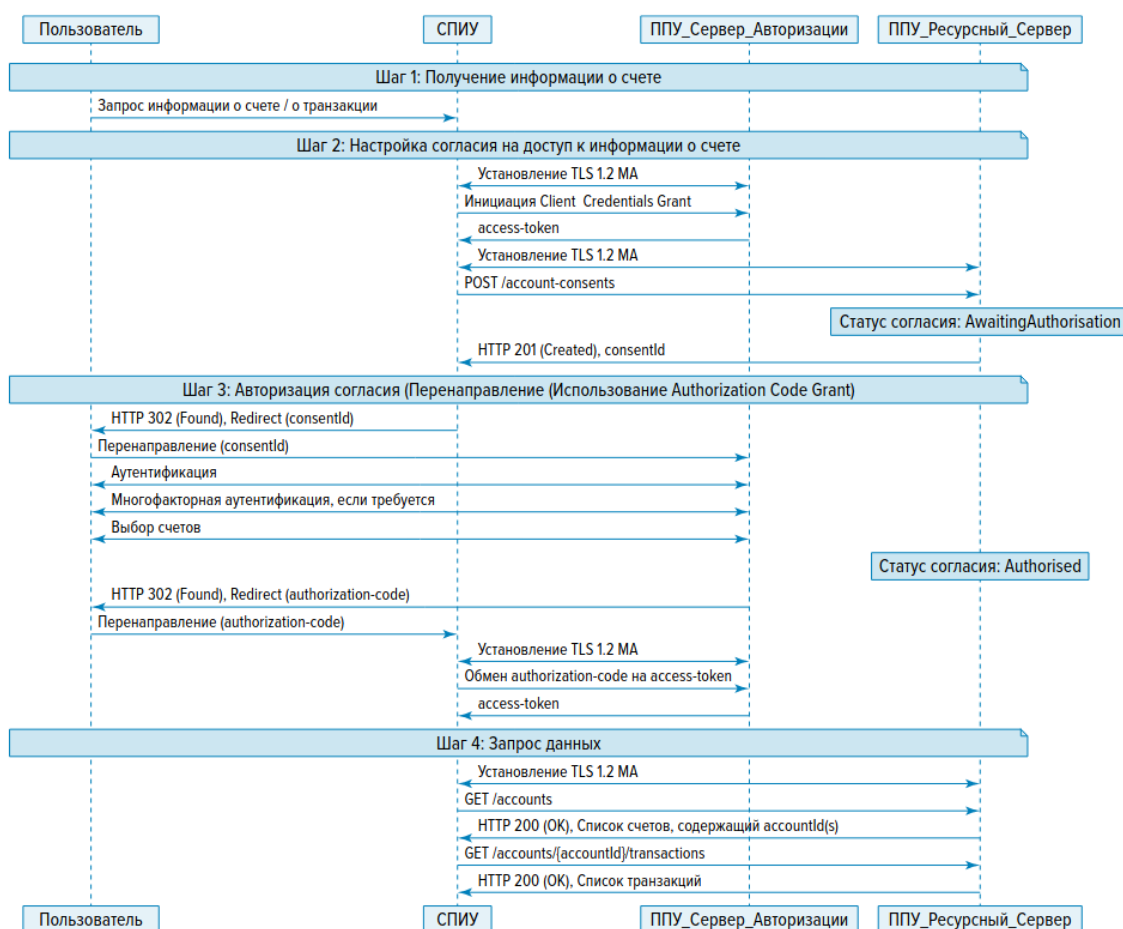


Рисунок 2. Диаграмма последовательности для получения информации о счете

После аутентификации клиента сторонний сервис может запросить у него доступ к счетам (в рамках доступа к счёту можно просматривать информацию о счёте, включая его номер, получать балансы, списки операций и генерировать выписки, а также просматривать информацию о компании).

Сторонний сервис отправляет запрос в Бланк на создание ресурса согласия с перечислением необходимых разрешений (чтение транзакций, чтение балансов и пр. – разобрано ниже), после чего клиент авторизует это согласие одним из доступных способов (поддерживается флоу с редиректом клиента на форму авторизации Бланка и флоу с использованием форм стороннего сервиса). Список счетов, к которым будет предоставлен доступ в рамках конкретного согласия, определяется самим клиентом в случае использования флоу с редиректом на форму Бланка. В случае реализации по флоу с использованием форм стороннего сервиса доступ ко всем счетам выдается фактически.

Создание ресурса согласия и его авторизация клиентом

Шаг 1. Создание ресурса согласия

Сторонний сервис отправляет запрос:

```
POST /api/accounts-service/open-banking/v1.0/account-consents
Content-Type: application/json
Authorization: Bearer ...

{
  "Data": {
    "Permissions": [
      "ReadAccountsBasic",
      "ReadAccountsDetail",
      "ReadBalances",
      "ReadTransactionsBasic",
      "ReadTransactionsCredits",
      "ReadTransactionsDebits",
      "ReadTransactionsDetail"
    ],
    "ExpirationDateTime": "2031-10-20T00:00:00Z",
    "TransactionFromDateTime": "2021-10-20T00:00:00Z",
    "TransactionToDateTime": "2031-10-20T00:00:00Z"
  }
}
```

Описание атрибутов:

| Значение | Описание |
|--|--|
| Permissions | Содержит необходимый стороннему сервису список разрешений |
| ExpirationDateTime | Передаётся время окончания действия согласия. Если поле не передано – согласие выдаётся на 10 лет с момента его создания |
| TransactionFromDateTime TransactionToDateTime | Можно указать период времени, за который будут доступны транзакции. Если этот период времени не передан, то сторонний сервис получит доступ к транзакциям клиента за всё время |

Возможные типы разрешений:

| Значение | Описание |
|-------------------------|--|
| ReadAccountsBasic | Просмотр базовой информации по счёту (тип, статус, валюта) |
| ReadAccountsDetail | Просмотр детальной информации по счёту (номер счёта) |
| ReadBalances | Просмотр балансов по счёту |
| ReadTransactionsBasic | Просмотр базовой информации по транзакциям |
| ReadTransactionsCredits | Просмотр кредитовых транзакций |
| ReadTransactionsDebits | Просмотр дебетовых транзакций |
| ReadTransactionsDetail | Просмотр детальной информации по транзакциям |

Возможные статусы TransactionStatusStaticType:

| Значение | Описание |
|----------|---|
| Booked | Забронировано |
| Pending | В ожидании, бронирование еще не произошло |

Успешный ответ возвращается в формате:

```
{
  "data": {
    "consentId": "77b7b7be-77c7-7d7f-a777-777aa77aa777",
    "creationDateTime": "2021-10-22T06:03:48.5397238Z",
    "status": "AwaitingAuthorisation",
    "statusUpdateDateTime": "2021-10-22T06:03:48.5397238Z",
    "permissions": [
      "ReadAccountsBasic",
      "ReadAccountsDetail",
      "ReadBalances",
      "ReadTransactionsBasic",
      "ReadTransactionsCredits",
      "ReadTransactionsDebits",
      "ReadTransactionsDetail"
    ],
    "expirationDateTime": "2031-10-20T00:00:00Z",
    "transactionFromDate": "2021-10-20T00:00:00Z",
    "transactionToDate": "2031-10-20T00:00:00Z"
  },
  "risk": {},
  "meta": {},
  "links": {
    "self": "https://m.blanc.ru/api/accounts-service/open-banking/v1.0/account-consents/77b7b7be-77c7-7d7f-a777-777aa77aa777"
  }
}
```


Проверить состояние согласия можно запросом:

```
GET /api/accounts-service/open-banking/v1.0/account-consents/{consentId}
Authorization: Bearer ...
```

Ответ вернётся в вышеописанном формате.

Согласие может находиться в одном из статусов (*status*):

| Значение | Описание |
|-----------------------|--|
| Authorised | Успешно авторизован, доступ к счетам предоставлен |
| AwaitingAuthorisation | Ожидает авторизации клиентом |
| Rejected | Клиент отклонил согласие |
| Revoked | Клиент отозвал согласие, доступ к счетам прекращён |

После создания ресурса согласия сторонний сервис должен запросить его авторизацию клиентом – доступны 2 способа, которые рассмотрены далее.

Шаг 2. Авторизация с редиректом на форму Бланка

Нужно направить пользователя на URL:

```
/payments/consent?ConsentId={consentId}&ReturnUrl={returnUrl}
```

Описание параметров:

| Значение | Описание |
|-----------|---|
| consentId | Идентификатор согласия, которое должно быть авторизовано |
| returnUrl | Адрес страницы, на которую необходимо вернуть пользователя после прохождения форм |

На странице Бланка пользователь сам выбирает, к каким счетам он хочет предоставить доступ. Пользователь имеет также возможность отказаться от авторизации согласия.

Шаг 2. Авторизация на формах партнёрского приложения

Авторизация в партнёрском приложении выполняется в два этапа – отправка одноразового кода клиенту и ввод кода клиентом.

Чтобы отправить код клиенту, сторонний сервис должен выполнить запрос:

```
POST /api/accounts-service/open-banking/v1.0/accounts-
consents/{consentId}/authorize/sendCode
Authorization: Bearer ...
```

Успешный ответ возвращается в формате:

```
{
  "value": {
    "status": "Sent",
    "userMessage": "Код был отправлен",
    "confirmToken": "...",
    "payload": {
      "refreshCodeTimeoutSeconds": 300
    }
  },
  "isSuccess": true,
  "error": null
}
```

Описание атрибутов:

| Значение | Описание |
|---------------------|--|
| confirmToken | Содержит краткосрочный токен, который необходимо передать при вводе одноразового кода клиентом |
| status | Содержит код результата отправки одноразового кода клиенту |
| InternalServerError | Произошла внутренняя ошибка (например, недоступен провайдер СМС) |
| Sent | Код успешно отправлен клиенту |
| Throttled | Слишком много попыток ввести код, необходимо подождать перед новой попыткой |
| Blocked | Отправка кодов клиенту заблокирована |

Чтобы передать введённый клиентом код, сторонний сервис должен выполнить запрос:

```
POST /api/accounts-service/open-banking/v1.0/accounts-
consents/{consentId}/authorize/verifyCode
Authorization: Bearer ...

{
  "Code": "1234",
  "ConfirmToken": "...
}
```

Успешный ответ возвращается в формате:

```
{
  "value": {
    "status": "Confirmed",
    "userMessage": "Код был введён верно",
    "payload": {}
  },
  "isSuccess": true,
  "error": null
}
```

Возможные коды результата ввода одноразового кода клиентом:

| Значение | Описание |
|-----------------------|--|
| InvalidCode | Введён неверный код |
| Confirmed | Введён верный код, согласие авторизовано |
| AttemptsExceeded | Использованы все попытки на ввод кода, необходимо запросить новый |
| Blocked | Ввод кодов клиентом заблокирован |
| InternalError | Произошла внутренняя ошибка |
| InvalidConfirmToken | Некорректный токен подтверждения |
| CodeNotSentOrOutdated | Код ещё не был отправлен или срок его действия истёк |
| InvalidRequest | Некорректный запрос (попытка ввести код, высланный для авторизации другого согласия) |

Шаг 3. Получение списка счетов, авторизованных в рамках согласия

Чтобы узнать, к каким счетам клиент предоставил доступ в рамках согласия, сторонний сервис должен выполнить запрос:

```
GET /api/accounts-service/open-banking/v1.0/account-
consents/{consentId}/accounts
Authorization: Bearer ...
```

Успешный ответ вернётся в формате:

```
{
  "data": {
    "account": [
      {
        "accountId": "77b7b7be-77c7-7d7f-a777-777aa77aa777",
        "status": "Enabled",
        "statusUpdateDateTime": "2021-10-19T06:23:00.277338",
        "currency": "RUB",
        "accountType": "Business",
        "accountSubType": "CurrentAccount",
        "accountDetails": {
          "schemeName": "RU.CBR.BBAN",
          "identification": "77777777777777777777"
        }
      }
    ]
  },
  "risk": {},
  "meta": {},
  "links": {
    "self": "https://m.blanc.ru/api/accounts-service/open-
banking/v1.0/account-consents/77b7b7be-77c7-7d7f-a777-777aa77aa777/accounts"
  }
}
```

Счёт может находиться в одном из статусов (*status*):

| Значение | Описание |
|--|--|
| Enabled | Доступен и используется клиентом |
| Disabled | Недоступен, временно или навсегда |
| Deleted | Закрыт и не может быть использован |
| Pending | Изменения по счёту ожидают одобрения (например, счёт зарезервирован перед открытием) |
| Тип счёта (<i>accountType</i>) | Всегда имеет значение Business |
| Подтип счёта (<i>accountSubType</i>) | На текущий момент всегда имеет значение CurrentAccount |

Просмотр информации по счёту

Чтобы просмотреть информацию по конкретному счёту, сторонний сервис должен выполнить запрос:

```
GET /api/accounts-service/open-banking/v1.0/accounts/{accountId}
Authorization: Bearer ...
```

Ответ возвращается в формате:

```
{
  "data": {
    "account": [
      {
        "accountId": "77b7b7be-77c7-7d7f-a777-777aa77aa777",
        "status": "Enabled",
        "statusUpdateDateTime": "2021-10-19T06:23:00.277338",
        "currency": "RUB",
        "accountType": "Business",
        "accountSubType": "CurrentAccount",
        "accountDetails": {
          "schemeName": "RU.CBR.BBAN",
          "identification": "77777777777777777777"
        }
      }
    ]
  },
  "risk": {},
  "meta": {},
  "links": {
    "self": "https://m.blanc.ru/api/accounts-service/open-banking/v1.0/accounts/77b7b7be-77c7-7d7f-a777-777aa77aa777"
  }
}
```

Просмотр балансов

Чтобы получить балансы по счёту, сторонний сервис должен выполнить запрос:

```
GET /api/accounts-service/open-banking/v1.0/accounts/{accountId}/balances
Authorization: Bearer ...
```

Успешный ответ возвращается в формате:

```
{
  "data": {
    "balance": [
      {
        "accountId": "77b7b7be-77c7-7d7f-a777-777aa77aa777",
        "creditDebitIndicator": "Debit",
        "type": "OpeningAvailable",
        "dateTime": "2021-10-22T07:40:45.9698676Z",
        "amount": {
          "amount": 98858.01,
          "currency": "RUB"
        }
      }
    ]
  },
  "risk": {},
  "meta": {},
  "links": {
    "self": "https://m.blanc.ru/api/processing-service/open-banking/v1.0/accounts/77b7b7be-77c7-7d7f-a777-777aa77aa777/balances"
  }
}
```

Тип баланса (`type`) может принимать несколько значений:

| Значение | Описание |
|------------------|--|
| ClosingAvailable | Конечный остаток суммы денег, которая находится в распоряжении владельца счета на указанную дату |
| ClosingBooked | Остаток по счету на конец предварительно согласованного отчетного периода. Это сумма начального зарегистрированного баланса в начале периода и всех записей, зачисленных на счет в течение предварительно согласованного отчетного периода |
| ClosingCleared | Конечный остаток суммы денег, которая очищается в указанную дату |

| | |
|------------------------|---|
| Expected | Баланс, состоящий из забронированных записей и отложенных позиций, известных на момент расчета, который прогнозирует остаток на конец дня, если все зарезервировано на счете и никакая другая запись не публикуется |
| OpeningAvailable | Начальный баланс суммы денег, которая находится в распоряжении владельца счета на указанную дату. На текущий момент возвращается только этот. |
| OpeningBooked | Баланс счета в начале отчетного периода. Он всегда равен балансу из предыдущего отчета |
| OpeningCleared | Начальный баланс, который очищается в указанную дату |
| PreviouslyClosedBooked | Остаток по счету за ранее закрытый отчетный период. Начальный зарегистрированный баланс на новый период равен этому балансу |

Отзыв согласия на доступ к счёту

Сторонний сервис выполняет запрос:

```
DELETE /api/accounts-service/open-banking/v1.0/accounts/{accountId}
Authorization: Bearer ...
```

Возвращается пустой ответ с кодом 204 No Content.

Отзыв согласия переводит его в статус Revoked.

Просмотр транзакций

Чтобы просмотреть операции по счёту, сторонний сервис должен выполнить запрос:

```
GET /api/transaction-service/open-
banking/v1.0/accounts/{accountId}/transactions?
pageSize=25&
pageIndex=0&
fromBookingDateTime=2021-10-01T00:00:00&
toBookingDateTime=2021-10-30T00:00:00&
fromStatusUpdateDateTime=2021-10-01T00:00:00&
toStatusUpdateDateTime=2021-10-01T00:00:00&
Authorization: Bearer ...
```

Описание атрибутов:

| Значение | Описание |
|--------------------------|---|
| pageSize | Отвечают за пейджинг (размер и индекс страницы соответственно) |
| pageIndex | |
| fromBookingDateTime | Задаётся период выборки операций (по дате создания транзакции) |
| toBookingDateTime | |
| fromStatusUpdateDateTime | При помощи параметров задаётся дополнительная фильтрация по времени обновления транзакций |
| toStatusUpdateDateTime | |

Успешный ответ возвращается в формате:

```
{
  "data": {
    "transaction": [
      {
        "accountId": "77b7b7be-77c7-7d7f-a777-777aa77aa777",
        "transactionId": "cfc7777b-77e6-4bad-a661-26b104fea991",
        "creditDebitIndicator": "Credit",
        "status": "AcceptedSettlementCompleted",
        "bookingDateTime": "2021-10-20T21:00:00",
        "valueDateTime": "2021-10-21T11:42:23",
        "transactionInformation": "тестовый платёж",
        "amount": {
          "amount": 100000.00,
          "currency": "RUB"
        },
        "chargeAmount": {
          "amount": 0.0,
          "currency": "RUB"
        },
        "creditorAgent": {
          "schemeName": "RU.CBR.BIK",
          "identification": "7777777777"
        },
        "creditorAccount": {
          "schemeName": "RU.CBR.BBAN",
          "identification": "77777777777777777777"
        },
        "debtorAgent": {
          "schemeName": "RU.CBR.BIK",
          "identification": "7777777777"
        },
        "debtorAccount": {
          "schemeName": "RU.CBR.BBAN",
          "identification": "77777777777777777777",
          "name": "ООО \"ЧЕНЬ ХОРОШО\""
        }
      }
    ],
    {
      "accountId": "77b7b7be-77c7-7d7f-a777-777aa77aa777",
```



```

"transactionId": "55f547cd-dabd-44c0-96e7-3a92ac85d11b",
"creditDebitIndicator": "Debit",
"status": "AcceptedSettlementCompleted",
"bookingDateTime": "2021-10-21T11:42:48.568893",
"valueDateTime": "2021-10-21T11:43:13",
"transactionInformation": "Покупка пакета платёжек",
"amount": {
  "amount": 200.00,
  "currency": "RUB"
},
"chargeAmount": {
  "amount": 0.0,
  "currency": "RUB"
},
"creditorAgent": {
  "schemeName": "RU.CBR.BIK",
  "identification": "7777777777"
},
"creditorAccount": {
  "schemeName": "RU.CBR.BBAN",
  "identification": null
},
"debtorAgent": {
  "schemeName": "RU.CBR.BIK",
  "identification": "7777777777"
},
"debtorAccount": {
  "schemeName": "RU.CBR.BBAN",
  "identification": "77777777777777777777"
}
},
{
"accountId": "77b7b7be-77c7-7d7f-a777-777aa77aa777",
"transactionId": "1cb6b631-28bc-436e-8ba1-94462cad2769",
"creditDebitIndicator": "Debit",
"status": "AcceptedSettlementCompleted",
"bookingDateTime": "2021-10-21T11:42:51.079373",
"valueDateTime": "2021-10-21T11:43:13",
"transactionInformation": "Покупка пакета переводов на
физлиц",
"amount": {
  "amount": 450.00,
  "currency": "RUB"
},
"chargeAmount": {
  "amount": 0.0,
  "currency": "RUB"
},
"creditorAgent": {
  "schemeName": "RU.CBR.BIK",
  "identification": "7777777777"
},
"creditorAccount": {
  "schemeName": "RU.CBR.BBAN",
  "identification": null
},
"debtorAgent": {

```

```

        "schemeName": "RU.CBR.BIK",
        "identification": "777777777"
    },
    "debtorAccount": {
        "schemeName": "RU.CBR.BBAN",
        "identification": "77777777777777777777"
    }
}
]
},
"risk": {},
"meta": {},
"links": {
    "self": "https://m.blanc.ru/api/transaction-service/open-
banking/v1.0/accounts/77b7b7be-77c7-7d7f-a777-
777aa77aa777%3FfromBookingDateTime=10/01/2021
00:00:00&toBookingDateTime=10/30/2021 00:00:00&pageSize=3&pageIndex=0"
}
}

```

Генерация и просмотр выписки

Выписки генерируются в асинхронном режиме, процесс состоит из двух шагов:

- запуск генерации выписки
- получение статуса выписки и самой выписки, если она готова

Шаг 1. Запуск генерации выписки

Чтобы запустить генерацию выписки, сторонний сервис должен выполнить запрос:

```

POST /api/transaction-service/open-banking/v1.0/statements
Content-Type: application/json
Authorization: Bearer ...

{
  "Data": {
    "Statement": {
      "AccountId": "77b7b7be-77c7-7d7f-a777-777aa77aa777",
      "FromBookingDateTime": "2021-01-01T00:00:00",
      "ToBookingDateTime": "2021-12-31T23:59:59"
    }
  }
}

```

В полях `FromBookingDateTime` и `ToBookingDateTime` передаются даты начала и окончания периода выписки. Даты передаются в UTC!

Ответ возвращается в формате:

```
{
  "data": {
    "statement": {
      "accountId": "77b7b7be-77c7-7d7f-a777-777aa77aa777",
      "statementId": "9303b823-47a6-4fcf-b125-dc8a2181e837",
      "fromBookingDateTime": "2021-01-01T00:00:00",
      "toBookingDateTime": "2021-12-31T23:59:59"
    }
  },
  "risk": {},
  "meta": {},
  "links": {
    "self": "https://m.blanc.ru/api/transaction-service/open-banking/v1.0/accounts/77b7b7be-77c7-7d7f-a777-777aa77aa777/statements/9303b823-47a6-4fcf-b125-dc8a2181e837"
  }
}
```

Шаг 2. Получение выписки

Чтобы получить выписку, сторонний сервис должен выполнить запрос:

```
GET /api/transaction-service/open-banking/v1.0/accounts/{accountId}/statements/{statementId}
Authorization: Bearer ...
```

Успешный ответ возвращается в формате:

```
{
  "data": {
    "statement": [
      {
        "accountId": "77b7b7be-77c7-7d7f-a777-777aa77aa777",
        "statementId": "9303b823-47a6-4fcf-b125-dc8a2181e837",
        "fromBookingDateTime": "2021-01-01T00:00:00",
        "toBookingDateTime": "2021-12-31T23:59:59",
        "creationDateTime": "2021-10-22T07:44:25.058603",
        "status": "Processing",
        "statusUpdateDateTime": "2021-10-22T07:44:30.568266"
      }
    ]
  },
  "risk": {},
  "meta": {},
  "links": {
    "self": "https://m.blanc.ru/api/transaction-service/open-banking/v1.0/accounts/77b7b7be-77c7-7d7f-a777-777aa77aa777/statements/9303b823-47a6-4fcf-b125-dc8a2181e837"
  }
}
```

Выписка может иметь следующие статусы (status):

| Значение | Описание |
|------------|---|
| Init | Выписка ещё не взята в обработку |
| Processing | Идёт процесс генерации выписки |
| Ready | Выписка готова |
| Failed | Генерация выписки закончилась с ошибкой |

Если выписка готова, то ответ возвращается в формате:

```
{
  "data": {
    "statement": [
      {
        "accountId": "77b7b7be-77c7-7d7f-a777-777aa77aa777",
        "statementId": "9303b823-47a6-4fcf-b125-dc8a2181e837",
        "fromBookingDateTime": "2021-01-01T00:00:00",
        "toBookingDateTime": "2021-12-31T23:59:59",
        "creationDateTime": "2021-10-22T07:44:25.058603",
        "status": "Ready",
        "statusUpdateDateTime": "2021-10-22T07:44:30.568266",
        "transaction": [
          {
            "transactionId": "cfc3903b-92e6-4bad-a661-
26b104fea991",
            "creditDebitIndicator": "Credit",
            "status": "AcceptedSettlementCompleted",
            "documentNumber": "100",
            "bookingDateTime": "2021-10-20T21:00:00",
            "valueDateTime": "2021-10-21T11:42:23",
            "description": "Тестовый платёж",
            "amount": {
              "amount": 100000.00,
              "currency": "RUB"
            },
            "debtorParty": {
              "inn": "7777777777",
              "name": "ООО \"САЛАТОВЫЙ БАНК\"",
              "kpp": "7777777777"
            },
            "debtorAgent": {
              "schemeName": "RU.CBR.BIK",
              "identification": "7777777777",
              "name": "ООО «Бланк банк»"
            },
            "debtorAccount": {
              "schemeName": "RU.CBR.BBAN",
              "identification": "77777777777777777777",
              "name": "ООО \"САЛАТОВЫЙ БАНК\""
            },
            "creditorParty": {},
            "creditorAgent": {
```

```

        "schemeName": "RU.CBR.BIK",
        "identification": "044525801",
        "name": "ООО «Бланк банк»"
    },
    "creditorAccount": {
        "schemeName": "RU.CBR.BBAN",
        "identification": "77777777777777777777"
    },
    "typeCode": "PaymentInstruction",
    "priority": 5
},
{
    "transactionId": "4a0d89f2-0b06-4f4a-88d7-
a4e50a206de8",
    "creditDebitIndicator": "Debit",
    "status": "AcceptedSettlementCompleted",
    "documentNumber": "7",
    "bookingDateTime": "2021-10-21T12:36:45.65181",
    "valueDateTime": "2021-10-21T12:37:48",
    "description": "Аванс по зарплате за Август
2015\nБез НДС.",
    "amount": {
        "amount": 10.00,
        "currency": "RUB"
    },
    "debtorParty": {},
    "debtorAgent": {
        "schemeName": "RU.CBR.BIK",
        "identification": "044525801",
        "name": "ООО «Бланк банк»"
    },
    "debtorAccount": {
        "schemeName": "RU.CBR.BBAN",
        "identification": "77777777777777777777"
    },
    "creditorParty": {
        "inn": "7777777777",
        "name": "Иванов Иван Иванович",
        "kpp": "7777777777"
    },
    "creditorAgent": {
        "schemeName": "RU.CBR.BIK",
        "identification": "7777777777",
        "name": "АО «САЛАТОВЫЙ БАНК»"
    },
    "creditorAccount": {
        "schemeName": "RU.CBR.BBAN",
        "identification": "77777777777777777777",
        "name": "Иванов Иван Иванович"
    },
    "typeCode": "PaymentInstruction",
    "priority": 5
}
],
"balance": [
    {
        "accountId": "77b7b7be-77c7-7d7f-a777-777aa77aa777",

```

```

        "creditDebitIndicator": "Debit",
        "type": "OpeningAvailable",
        "dateTime": "2021-01-01T00:00:00",
        "amount": {
            "amount": 0.0,
            "currency": "RUB"
        }
    },
    {
        "accountId": "77b7b7be-77c7-7d7f-a777-777aa77aa777",
        "creditDebitIndicator": "Debit",
        "type": "ClosingAvailable",
        "dateTime": "2021-12-31T23:59:59",
        "amount": {
            "amount": 98858.01,
            "currency": "RUB"
        }
    }
]
}
},
"risk": {},
"meta": {},
"links": {
    "self": "https://m.blanc.ru/api/transaction-service/open-
banking/v1.0/accounts/77b7b7be-77c7-7d7f-a777-
777aa77aa777/statements/9303b823-47a6-4fcf-b125-dc8a2181e837"
}
}

```

Описание атрибутов:

| Значение | Описание |
|-----------------------------|---|
| transaction | Передаётся список операций |
| balance | Балансы |
| TransactionStatusStaticType | Статус записи транзакции. |
| Booked | Забронировано |
| Pending | В ожидании, бронирование еще не произошло |

Тип баланса (type) может принимать несколько значений:

| Значение | Описание |
|------------------|--|
| ClosingAvailable | Конечный остаток суммы денег, которая находится в распоряжении владельца счета на указанную дату |
| ClosingBooked | Остаток по счету на конец предварительно согласованного отчетного периода. Это сумма |

| | |
|------------------------|---|
| | начального зарегистрированного баланса в начале периода и всех записей, зачисленных на счет в течение предварительно согласованного отчетного периода |
| ClosingCleared | Конечный остаток суммы денег, которая очищается в указанную дату |
| Expected | Баланс, состоящий из забронированных записей и отложенных позиций, известных на момент расчета, который прогнозирует остаток на конец дня, если все зарезервировано на счете и никакая другая запись не публикуется |
| OpeningAvailable | Начальный баланс суммы денег, которая находится в распоряжении владельца счета на указанную дату. На текущий момент возвращается только этот. |
| OpeningBooked | Баланс счета в начале отчетного периода. Он всегда равен балансу из предыдущего отчета |
| OpeningCleared | Начальный баланс, который очищается в указанную дату |
| PreviouslyClosedBooked | Остаток по счету за ранее закрытый отчетный период. Начальный зарегистрированный баланс на новый период равен этому балансу |

Просмотр информации о компании

Чтобы получить информацию о компании, которой принадлежит счёт клиента, сторонний сервис должен выполнить запрос:

```
GET /api/accounts-service/open-banking/v1.0/accounts/{accountId}/company-
details
Authorization: Bearer ...
```

Успешный ответ возвращается в формате:

```
{
  "data": {
    "inn": "470424061524",
    "fullName": "Индивидуальный предприниматель Балан Валентин",
    "shortName": "ИП Балан Валентин",
    "mainOkved": "49.41",
    "registrationAddressLine": "г Уфа, ул Дагестанская, д 1",
    "registrationDate": "2020-10-20T00:00:00"
  },
  "risk": {},
  "meta": {},
  "links": {
    "self": "https://m.blanc.ru/api/accounts-service/open-
banking/v1.0/accounts/77b7b7be-77c7-7d7f-a777-777aa77aa777/company-details"
  }
}
```


Создание и отправка платежей

6.2.1.2. Диаграмма последовательности

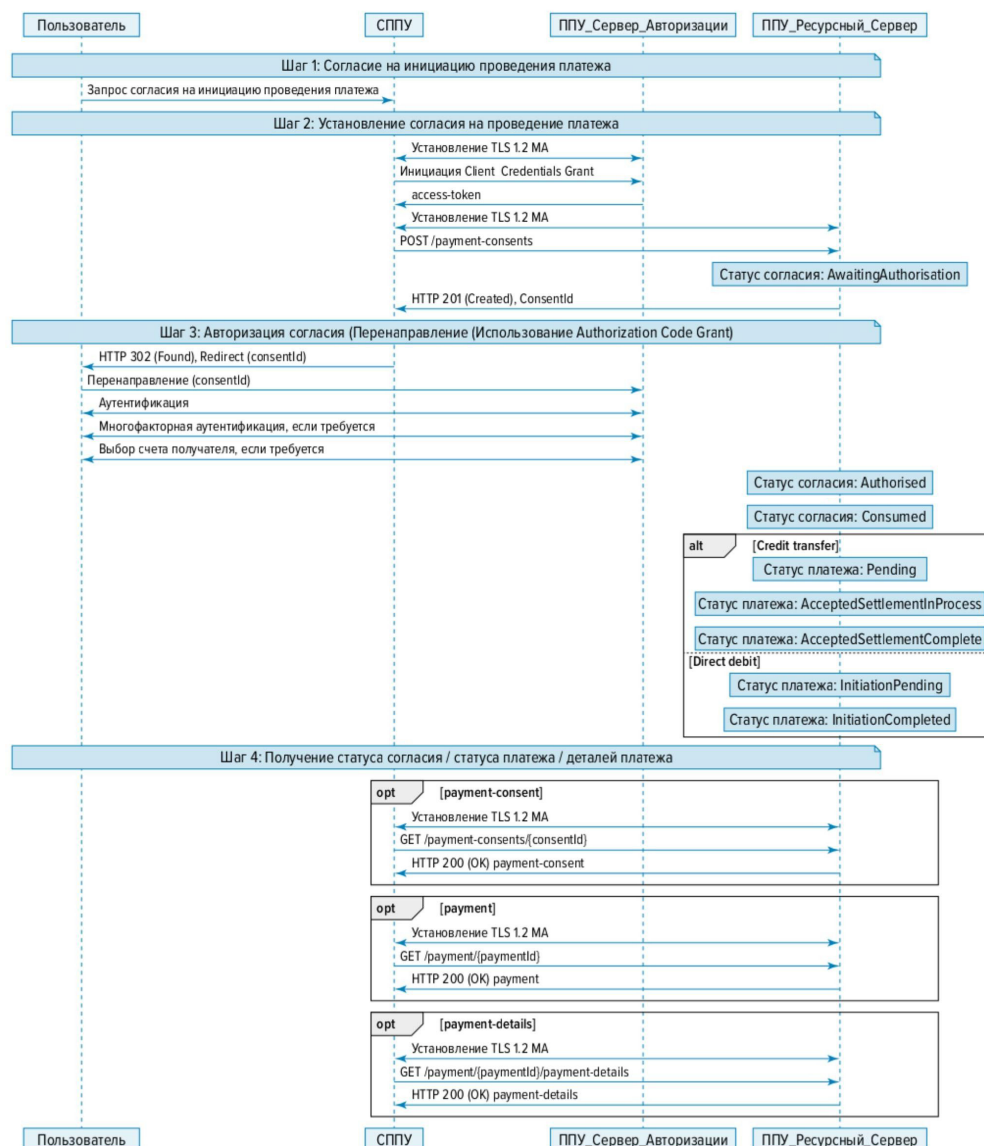


Рисунок 2. Диаграмма последовательности для инициирования платежа

После аутентификации клиента сторонний сервис может создавать платежи и запрашивать их подписание клиентом.

Сторонний сервис отправляет запрос в Бланк на создание ресурса согласия с указанием необходимых данных платежа (счёт получателя, сумма, назначение и пр.), после чего клиент авторизует это согласие одним из доступных способов (поддерживается флоу с редиректом клиента на форму авторизации Бланка и флоу с использованием форм стороннего сервиса). Счёт списания указывается сторонним сервисом.

Создание одиночного платежа и его подписание клиентом

Шаг 1. Создание ресурса согласия (перевод бизнесу)

Сторонний сервис отправляет запрос:

```
POST /api/transaction-service/open-banking/v1.0/payment-consents
Content-Type: application/json
Authorization: Bearer ...
```

```
{
  "Data": {
    "Initiation": {
      "InstructionIdentification": "instruction-id-100",
      "EndToEndIdentification": "end-to-end-id-100",
      "InstructedAmount": {
        "Amount": "111.99",
        "Currency": "RUB"
      },
    },
    "DebtorAccount": {
      "SchemeName": "RU.CBR.BBAN",
      "Identification": "77777777777777777777",
      "Name": null
    },
    "CreditorParty": {
      "Name": "ООО Рога и Копыта",
      "Inn": "7777777777",
      "Kpp": "7777777777"
    },
    "CreditorAgent": {
      "SchemeName": "RU.CBR.BIK",
      "Identification": "7777777777"
    },
    "CreditorAccount": {
      "SchemeName": "RU.CBR.BBAN",
      "Identification": "77777777777777777777",
      "Name": "АО КАКОЙ-НИБУДЬ БАНК"
    },
    "RemittanceInformation": {
      "Unstructured": "Тестовое назначение платежа"
    },
    "VatType": "Vat10",
    "DocumentNumber": 42,
    "Priority": 5
  },
  "Authorisation": {
    "AuthorisationType": "Single"
  }
},
"Risk": {}
}
```

Описание атрибутов:

| Значение | Описание |
|------------------------------------|--|
| RemittanceInformation.Unstructured | Передаётся назначение платежа |
| VatType | Передаётся тип НДС |
| DocumentNumber | Передаётся номер платёжного документа. Поле необязательное: если не передать значение, то номер будет назначен банком автоматически |
| Priority | Передаётся приоритет платёжного документа. Поле необязательное: если не передать значение, то будет задан приоритет по умолчанию (5) |

Возможные значения VatType:

| Значение | Описание |
|------------|-------------|
| WithoutVat | без НДС |
| Vat10 | НДС 10% |
| Vat20 | НДС 20% |
| Other | прочие типы |

Ответ возвращается в формате:

```

{
  "data": {
    "initiation": {
      "instructionIdentification": "instruction-id-100",
      "endToEndIdentification": "end-to-end-id-100",
      "instructedAmount": {
        "amount": 111.99,
        "currency": "RUB"
      },
      "debtorAccount": {
        "schemeName": "RU.CBR.BBAN",
        "identification": "77777777777777777777"
      },
      "creditorParty": {
        "name": "ООО Рога и Копыта",
        "inn": "7777777777",
        "kpp": "777777777"
      },
      "creditorAgent": {
        "schemeName": "RU.CBR.BIK",
        "identification": "777777777"
      },
      "creditorAccount": {
        "schemeName": "RU.CBR.BBAN",

```

```

        "identification": "77777777777777777777",
        "name": "АО Крепкая сталь"
    },
    "remittanceInformation": {
        "unstructured": "Тестовое назначение платежа"
    },
    "vatType": "Vat10",
    "documentNumber": 42,
    "priority": 5
},
"authorisation": {
    "authorisationType": "Single"
},
"consentId": "30ca3036-2b88-40df-8bc5-f54c8170eadf",
"creationDateTime": "2021-10-22T10:01:53.8531983Z",
"status": "AwaitingAuthorisation",
"statusUpdateDateTime": "2021-10-22T10:01:53.8531983Z",
"charges": [
    {
        "chargeBearer": "DEBT",
        "type": "COMM",
        "amount": {
            "amount": 0.0,
            "currency": "RUB"
        }
    }
]
},
"risk": {},
"meta": {},
"links": {
    "self": "https://m.blanc.ru/api/transaction-service/open-
banking/v1.0/payment-consents/30ca3036-2b88-40df-8bc5-f54c8170eadf"
}
}

```

Описание атрибутов:

| Значение | Описание |
|--------------|---|
| charges | Передают предварительно рассчитанные комиссии. Если массив пустой, рассчитать комиссию не удалось. Нулевая комиссия передаётся с amount: 0.0. |
| chargeBearer | Указывает на то, кто платит комиссию – всегда возвращается значение DEBT (плательщик) |
| type | Определяет тип комиссии – всегда возвращается значение COMM (комиссия за оказанные банковские услуги) |

Проверить состояние согласия можно запросом:

```
GET /api/transaction-service/open-banking/v1.0/payment-consents/{consentId}
Authorization: Bearer ...
```

Ответ вернётся в вышеописанном формате.

Согласие может находиться в одном из статусов (status):

| Значение | Описание |
|-----------------------|---|
| Authorised | Успешно авторизован, платёж будет передан в обработку в ближайшее время |
| AwaitingAuthorisation | Ожидает авторизации клиентом |
| Rejected | Клиент отклонил согласие |
| Consumed | Согласие было задействовано, т. е. платёж в рамках согласия был передан в обработку |
| AwaitingAuthorisation | При получении согласия в статусе будет автоматически актуализирована комиссия |

В остальных статусах в ответе придёт пустой список charges.

Шаг 1. Создание ресурса согласия (перевод физ. лицу)

Сторонний сервис отправляет запрос:

```
POST /api/transaction-service/open-banking/v1.0/payment-consents
Content-Type: application/json
Authorization: Bearer ...
```

```
{
  "Data": {
    "Initiation": {
      "InstructionIdentification": "instruction-id-100",
      "EndToEndIdentification": "end-to-end-id-100",
      "InstructedAmount": {
        "Amount": "111.99",
        "Currency": "RUB"
      },
    },
    "DebtorAccount": {
      "SchemeName": "RU.CBR.BBAN",
      "Identification": "77777777777777777777",
      "Name": null
    },
    "CreditorParty": {
      "Name": "Кудрявцев Максим Андреевич"
    },
    "CreditorAgent": {
```

```

        "SchemeName": "RU.CBR.BIK",
        "Identification": "777777777"
    },
    "CreditorAccount": {
        "SchemeName": "RU.CBR.BBAN",
        "Identification": "77777777777777777777",
        "Name": "Альфабанк"
    },
    "RemittanceInformation": {
        "Unstructured": "Тестовое назначение платежа"
    }
},
"Authorisation": {
    "AuthorisationType": "Single"
}
},
"Risk": {}
}

```

Шаг 1. Создание ресурса согласия (налоговый платёж)

Сторонний сервис отправляет запрос:

```

POST /api/transaction-service/open-banking/v1.0/payment-consents
Content-Type: application/json
Authorization: Bearer ...

{
  "Data": {
    "Initiation": {
      "InstructionIdentification": "instruction-id-100",
      "EndToEndIdentification": "end-to-end-id-100",
      "InstructedAmount": {
        "Amount": 10,
        "Currency": "RUB"
      },
      "DebtorAccount": {
        "SchemeName": "RU.CBR.BBAN",
        "Identification": "77777777777777777777",
        "Name": "Иванова Галина Ивановна"
      },
      "CreditorParty": {
        "Name": "ИФНС РОССИИ ПО Г.АРХАНГЕЛЬСКУ",
        "Inn": "7777777777",
        "Kpp": "7777777777"
      },
      "CreditorAgent": {
        "SchemeName": "RU.CBR.BIK",
        "Identification": "111111111"
      },
      "CreditorAccount": {
        "SchemeName": "RU.CBR.BBAN",
        "Identification": "77777777777777777777",
        "Name": null
      }
    }
  }
}

```

```

    "RemittanceInformation": {
      "Unstructured": "Тестовый платёж"
    },
    "TaxInformation": {
      "uin": "0",
      "payerStatus": "13",
      "kbk": "66666666666666666666",
      "oktmo": "66666666",
      "paymentReason": "0",
      "period": "ГД.00.2020",
      "documentNumber": "0",
      "documentDate": "0"
    }
  },
  "Authorisation": {
    "AuthorisationType": "Single"
  }
},
"Risk": {}
}

```

Шаг 2. Авторизация с редиректом на форму Бланка

Нужно направить пользователя на URL:

```

/payments/confirm?ConsentId={consentId}&ReturnUrl={returnUrl}

```

Описание атрибутов:

| Значение | Описание |
|-----------|---|
| consentId | Идентификатор согласия, которое должно быть авторизовано |
| returnUrl | Адрес страницы, на которую необходимо вернуть пользователя после прохождения форм |

На странице Бланка пользователь подписывает платёж. Пользователь имеет также возможность отказаться от проведения платежа, в таком случае он будет переведен в статус Rejected.

Шаг 2. Авторизация на формах партнёрского приложения

Авторизация в партнёрском приложении выполняется аналогично авторизации согласия на доступ к счёту: в два этапа – отправка одноразового кода клиенту и ввод кода клиентом.

Чтобы отправить код клиенту, сторонний сервис должен выполнить запрос:

```
POST /api/transaction-service/open-banking/v1.0/payment-
consents/{consentId}/authorize/sendCode
Authorization: Bearer ...
```

Успешный ответ возвращается в формате:

```
{
  "value": {
    "status": "Sent",
    "userMessage": "Код был отправлен",
    "confirmToken": "53cdec52-868a-4d6b-af8f-095348d5be4a",
    "payload": {
      "refreshCodeTimeoutSeconds": 300
    }
  },
  "isSuccess": true,
  "error": null
}
```

Чтобы передать введённый клиентом код, сторонний сервис должен выполнить запрос:

```
POST /api/transaction-service/open-banking/v1.0/payment-
consents/{consentId}/authorize/verifyCode
Authorization: Bearer ...

{
  "Code": "1234",
  "ConfirmToken": "53cdec52-868a-4d6b-af8f-095348d5be4a"
}
```

Ответ возвращается в формате:

```
{
  "value": {
    "status": "Confirmed",
    "userMessage": "Согласие успешно подтверждено, платёж передан в
обработку",
    "payload": {
      "paymentId": "8dfe2dc7-faf5-44a9-ae5a-be7df349a640"
    }
  },
  "isSuccess": true,
  "error": null
}
```


В `payload.paymentId` возвращается идентификатор созданного платежа.

Если на счету клиента недостаточно средств для совершения платежа, то вернётся такой ответ:

```
{
  "value": null,
  "isSuccess": false,
  "error": {
    "type": "InsufficientFunds",
    "message": "Недостаточно средств"
  }
}
```

Согласие на платёж при этом будет переведено в статус `Rejected`.

Просмотр информации по одиночному платежу

Чтобы получить статус платежа, созданного в рамках согласия, сторонний сервис выполняет запрос:

```
GET /api/transaction-service/open-banking/v1.0/payments/{paymentId}
Authorization: Bearer ...
```

Ответ возвращается в формате:

```
{
  "data": {
    "paymentId": "8dfe2dc7-faf5-44a9-ae5a-be7df349a640",
    "consentId": "1cd78672-972b-428b-8129-61904d85eae6",
    "creationDateTime": "2021-10-22T10:14:47.656532",
    "status": "AcceptedSettlementInProgress",
    "statusUpdateDateTime": "2021-10-22T10:14:47.553445",
    "charges": [],
    "initiation": {
      "instructionIdentification": "instruction-id-100",
      "endToEndIdentification": "end-to-end-id-100",
      "instructedAmount": {
        "amount": 111.99,
        "currency": "RUB"
      },
      "debtorAccount": {
        "schemeName": "RU.CBR.BBAN",
        "identification": "77777777777777777777"
      },
      "creditorParty": {
        "name": "ООО Рога и Копыта",
        "inn": "7777777777",
        "kpp": "7777777777"
      },
      "creditorAgent": {
        "schemeName": "RU.CBR.BIK",

```

```

        "identification": "777777777"
    },
    "creditorAccount": {
        "schemeName": "RU.CBR.BBAN",
        "identification": "77777777777777777777",
        "name": "АО Крепкая сталь"
    },
    "remittanceInformation": {
        "unstructured": "Тестовое назначение платежа"
    },
    "charges": [
        {
            "chargeBearer": "DEBT",
            "type": "COMM",
            "amount": {
                "amount": 0.0,
                "currency": "RUB"
            }
        }
    ]
},
"risk": {},
"meta": {},
"links": {
    "self": "https://m.blanc.ru/api/transaction-service/open-banking/v1.0/payments/8dfe2dc7-faf5-44a9-ae5a-be7df349a640"
}
}

```

Платёж может находиться в одном из следующих статусов (status):

| Значение | Описание |
|-----------------------------------|---|
| AcceptedSettlementCompleted | Платёж исполнен (средства списаны со счёта плательщика) |
| AcceptedSettlementInProgress | Платёж находится в обработке |
| Pending | Платёж ожидает передачи в обработку |
| Rejected | Платёж отклонён |
| AcceptedWithoutPosting | Платёж принят без зачисления на счёт получателя |
| AcceptedCreditSettlementCompleted | Платёж исполнен (средства поступили на счёт получателя) |
| charges | Возвращается фактически удержанная комиссия |

Альтернативно, сторонний сервис может выполнить запрос:

```

GET /api/transaction-service/open-banking/v1.0/payments/{paymentId}/
payment-details
Authorization: Bearer ...

```

Ответ возвращается в формате:

```
{
  "data": {
    "paymentTransactionId": "ae7611a6-f197-4718-98aa-ebb4456e23c6",
    "status": "ACSC",
    "statusUpdateDateTime": "2021-10-22T10:16:37.957822"
  },
  "risk": {},
  "meta": {},
  "links": {
    "self": "https://m.blanc.ru/api/transaction-service/open-banking/v1.0/payments/8dfe2dc7-faf5-44a9-ae5a-be7df349a640/payment-details"
  }
}
```

В таком случае status принимает значения:

| Значение | Описание |
|----------|---|
| ACCC | соответствует статусу AcceptedCreditSettlementCompleted |
| ACSP | AcceptedSettlementInProgress |
| ACSC | AcceptedSettlementCompleted |
| ACWP | AcceptedWithoutPosting |
| PDNG | Pending |
| RJCT | Rejected |

Расчёт комиссии по параметрам платежа

Чтобы рассчитать комиссию для платежа, сторонний сервис выполняет запрос:

```
POST /api/transaction-service/open-banking/v1.0/payments/charges
Content-Type: application/json
Authorization:
Bearer ...

{
  "Data": {
    "InstructedAmount": {
      "Amount": "1100001.99",
      "Currency": "RUB"
    },
    "DebtorAccount": {
      "SchemeName": "RU.CBR.BBAN",
      "Identification": "77777777777777777777"
    },
    "CreditorAgent": {
      "SchemeName": "RU.CBR.BIK",

```

```

        "Identification": "777777777"
    },
    "CreditorAccount": {
        "SchemeName": "RU.CBR.BBAN",
        "Identification": "77777777777777777777"
    }
},
"Risk": {}
}

```

Ответ возвращается в формате:

```

{
  "data": {
    "charges": [
      {
        "chargeBearer": "DEBT",
        "type": "COMM",
        "amount": {
          "amount": 50.00,
          "currency": "RUB"
        }
      }
    ]
  },
  "risk": {},
  "meta": {},
  "links": {
    "self": ""
  }
}

```

Если комиссии не будут удержаны, то вернётся нулевая сумма.